

8 次格子グラフによる 2 次元画像の解像度低減化

神藤悠希^{†1} 穴田浩一^{†2} 夜久竹夫^{†1}

不均一な矩形分割を 8 次格子グラフにより表現すると効率よく 2 次元画像の解像度を低減することができる。本論文は、はじめに“octgrid”と呼ばれる不均一型矩形分割のための 8 次格子グラフに基づいたデータ構造を解説する。そして octgrid の技法を応用して、2 次元白黒画像を不均一な矩形分割で表現するための解像度低減化アルゴリズムを提案する。さらに、例を用いて我々の方法とヒルベルト走査法との比較を行い、いくつかの例で我々の方法がヒルベルト走査法より矩形数を 50%以下に低減することを示す。

Reduction of Resolution for 2D Images by an Octal Grid Graph Representation Model

YUKI SHINDO^{†1} KOICHI ANADA^{†2}
TAKEO YAKU^{†1}

We can reduce efficiently the resolution of a 2D image by expressing heterogeneous rectangular dissections in octal grid graphs. In this paper, we first survey a data structure called an “octgrid” which is a type of octal grid graphs for representing heterogeneous rectangular dissections such as 2D images. Then, we apply the technique of an octgrid, and propose algorithms for reduction of a resolution. Furthermore, we compare by examples our method with the Hilbert scanning, and show that our method reduces the rectangular number by less than 50% than the Hilbert scanning does, in several examples.

1. はじめに

本研究では白黒のラスター画像を対象とした、表示に適したデータ構造について扱う[8, 9].

2 次元画像を均一な矩形分割で表現する場合、一部の解像度を高くするために、画像全体の解像度を高くする必要があるのである。そのため、ディスプレイなどの端末に表示するために大きな計算時間を要し、作業効率の低下などの問題があげられる。

先行研究として矩形分割の表現のためのいくつかのデータ構造が知られている。不均一な矩形分割を表現可能なデータ構造として、Finkel & Bentley は木構造の一種である 4 分木を導入[2]し、Kozminsky & Kinnen は矩形双対グラフの性質を説明した[6]。夜久は、octgrid と呼ばれる不均一型矩形分割のための 8 次格子グラフに基づいたデータ構造を導入した[10]。octgrid はいくつかの変形に関して矩形双対グラフのデータ構造より計算時間が速いことが知られている。また Hilbert は、空間充填曲線の一つであるヒルベルト曲線を提示した[3]。2 次元画像の解像度を減らす方法として、ヒルベルト曲線に沿って走査を行うヒルベルト走査法を用いる方法は有名な方法の一つである。しかし、ヒルベルト走査法を用いる解像度低減化では走査順序による制約があり、低減化が不十分な場合がある。一方、octgrid の技

法を用いると柔軟な走査ができて 2 次元画像の解像度を効率よく低減することが可能である。そこで、本論文では、3D 地形図の解像度低減化アルゴリズム[1]を応用し、2 次元画像を不均一な矩形分割で表現する方法を示す。さらに、いくつかの例を用いて、我々の方法により低減化された 2 次元画像と従来方法であるヒルベルト走査法を用いる方法により低減化された 2 次元画像との解像度の比較を行う。

2 節では、準備として矩形分割、8 次格子グラフモデル、ヒルベルト走査法について解説を行う。3 節では、矩形分割内の合併操作を行うアルゴリズム[5]を紹介し、そのアルゴリズムを用いて 2 次元画像の解像度低減化アルゴリズムを提案する。さらに、4 節では、例を用いて、我々の方法によって低減化された 2 次元画像とヒルベルト走査法によって低減化された 2 次元画像の解像度の比較を行い、いくつかの 2 次元画像の例で我々の方法がヒルベルト走査法を用いる方法より解像度を 50%以下に低減することを示す。最後に、5 節で結論を述べる。

2. 準備

2.1 矩形分割

矩形分割は、ある平面上の矩形領域を共通部分のない幾つかの矩形で分割することである。その矩形は 2 種類に分類し、分割されていない矩形をセルと呼び、行もしくは列を表すための周囲の矩形を周辺セルと呼ぶ。また、一つ一つのセルを形成している線を壁という。一つのセルに対して、上に位置する壁を北壁、下に位置する壁を南壁、左に位置する壁を西壁、右に位置する壁を東壁という。その壁

^{†1} 日本大学
Nihon University
^{†2} 早稲田大学高等学院
Waseda University Senior High School

ごとに座標値が付随し、北壁・南壁にはY座標の値を持ち、東壁・西壁にはX座標の値を持つ。なお、周辺セルは、 x 方向・ y 方向どちらかの幅を持たないと定義する。

図1は矩形分割の例である。太線の矩形一つがセルであり、太線で形成している矩形の周りに存在する矩形が行・列を表す周辺セルである。数値は座標値で、例えば、セル c の北壁、南壁、東壁、西壁はそれぞれ0, 1, 2, 1である。

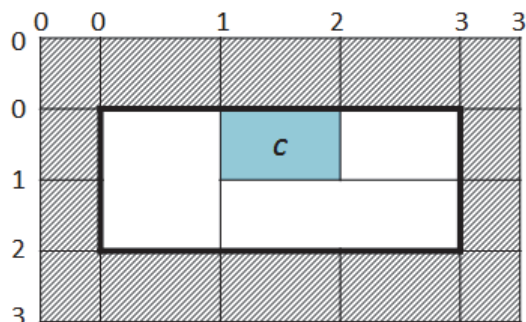


図1 矩形分割の例

2.2 8次格子グラフモデル

不均一矩形分割を表すグラフとして、夜久は octgrid と呼ばれる 8 次格子グラフに基づいた構造を導入した[10]. octgrid の頂点はセルもしくは周辺セルを表す。また、octgrid の辺は壁の位置関係を表し、同じ壁を共有する最も近い 2 つのセルに対応する 2 頂点を辺で結ぶ。以下のように octgrid を定義する。

定義 2.2.1

D を矩形分割とする。 D に対応する octgrid は、多重無向グラフ $G_D=(V_D, L, E_D, A_D, \alpha_D)$ である。

ただし、

V_D : D のセルを表す頂点集合。

L : 以下の①～④で定まるラベルの集合、 $L=\{enw, esw, eew, eww\}$ とする。

E_D : $E_D \subseteq V_D \times L \times V_D$ はラベル付きの無向辺の集合である、ただし、頂点 v_c と頂点 v_d 、ラベル l に対し、 $[v_c, l, v_d]$ と表し、次の①～④によって定義する。

- ① $nw(c)=nw(d)$ (セル c, d の北壁が共通) で、 c, d が最も近い位置にある場合、辺 $[v_c, enw, v_d]$ は E_D に属し、**北壁辺**と呼ぶ。
- ② $sw(c)=sw(d)$ (セル c, d の南壁が共通) で、 c, d が最も近い位置にある場合、辺 $[v_c, esw, v_d]$ は E_D に属し、**南壁辺**と呼ぶ。
- ③ $ew(c)=ew(d)$ (セル c, d の東壁が共通) で、 c, d が最も近い位置にある場合、辺 $[v_c, eew, v_d]$ は E_D に属し、**東壁辺**と呼ぶ。
- ④ $ww(c)=ww(d)$ (セル c, d の西壁が共通) で、 c, d が最も近い位置にある場合、辺 $[v_c, eww, v_d]$ は E_D に属し、**西壁辺**と呼ぶ。

A_D : 属性集合 $\subseteq R^4$ (セルの左上隅の xy 座標と、幅と高さの集まり)

α_D : 頂点に属性を持たせる写像、 $\alpha_D=(nw(c), sw(c), ew(c), ww(c), color(c))$ とする。

□

定義 2.2.2

グラフ G が octgrid であるとは、 G がある矩形分割に対応する octgrid となっていることである。

□

octgrid の最大の次数は 8 である。octgrid の上のいくつかの変形アルゴリズムは、矩形双対グラフのようなよく知られているデータ構造の上の対応するアルゴリズムより速いことが知られている。図2は図1に対応する octgrid を示している。

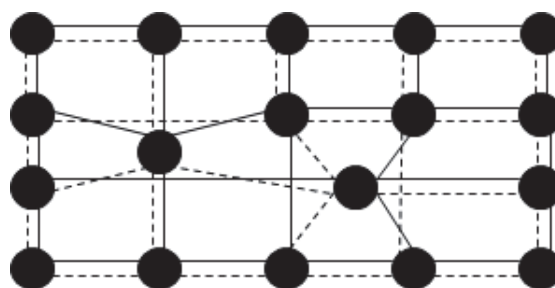


図2 図1に対応する octgrid

2.3 ヒルベルト走査法

Hilbert は 1891 年にフラクタル図形の一つで、空間を覆い尽くす空間充填曲線の一つである“ヒルベルト曲線”を提示した[3]. 図3に示されるように、この曲線は2次元平面のタイルを隙間なく通過する。

さらに、このヒルベルト曲線は2次元空間内のデータを走査するためによく用いられている[4]. その走査方法は、“ヒルベルト走査法”と呼ばれ、画像処理分野において画像圧縮などに広く適用されている。

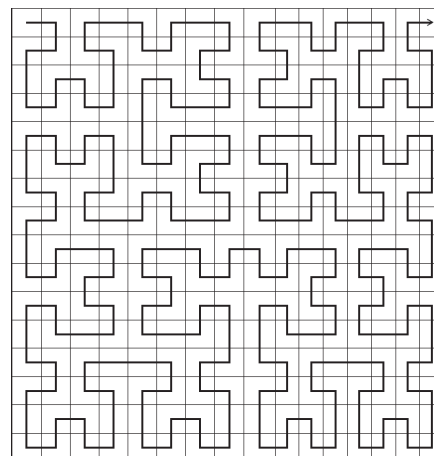


図3 ヒルベルト曲線の例

3. 2次元画像の表現

本節では、はじめに、矩形分割のセルの合併操作を行うアルゴリズムを紹介する。次に、2次元画像に対応するoctgridのセルを2次元画像のピクセルに対応させ、図形の特徴を保ったままセルの総数を減らすための2つのアルゴリズムを提案する。

- ① ヒルベルト曲線、垂直方向、水平方向の順に走査しセルを合併していく。
- ② ヒルベルト曲線、水平方向、垂直方向の順に走査しセルを合併していく。

また、この方法は、柔軟な走査が行えるというoctgridの利点を活かしている。

3.1 セルの合併[5]

矩形分割のセルの合併操作を行うアルゴリズムである“UnifyCell” [5]を用いる。“UnifyCell”は $ew(c)=ew(d)$, $ww(c)=ww(d)$, $nw(c)=nw(d)$, $sw(c)=sw(d)$, $color(c)=color(d)$ または, $nw(c)=nw(d)$, $sw(c)=sw(d)$, $ew(c)=ew(d)$, $color(c)=color(d)$ の場合に矩形分割に対応しているoctgridの“2つ並んだ”頂点を合併する。このアルゴリズムを用いることによって、矩形分割内のセルの数を減少させることができる。

Algorithm UnifyCell(G_D, v_c, v_d, G_E)

入力 G_D : 矩形分割 D に対応する octgrid.

v_c : G_D におけるセル c に対応する頂点.

v_d : G_D におけるセル d に対応する頂点.

出力 G_E : セル $c \cdot d$ が合併しセル c となった矩形分割 E に対応する octgrid.

方法

1. 初期化 $G_E \leftarrow G_D$.
2. v_c から出ている西壁辺・東壁辺を繋ぎ直す。
3. v_d から出ている西壁辺・東壁辺を繋ぎ直す。
4. v_d から出ている西壁辺・東壁辺を削除する。
5. v_c から出ている北壁辺・南壁辺を繋ぎ直す。
6. v_d から出ている北壁辺・南壁辺を繋ぎ直す。
7. v_d から出ている北壁辺・南壁辺を削除する。
8. v_d を削除する。

図4は“UnifyCell”の例である。

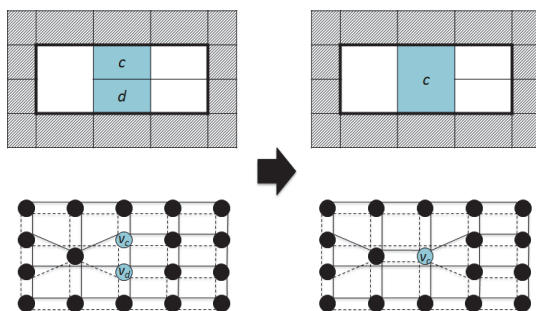


図4 “UnifyCell”によるセルの合併の例

3.2 解像度低減化

我々は3D地形図の解像度低減化アルゴリズム[1]を応用し、2次元画像の解像度低減を行う。

Algorithm 1 (ヒルベルト走査法を用いた合併→垂直方向合併→水平方向合併)

入力 G_D : 原画像の矩形分割 D に対応する octgrid ($2^n \times 2^n$ サイズ).

出力 G_E : 解像度が低減化された矩形分割 E に対応する octgrid.

方法

1. 初期化 $G_E \leftarrow G_D$.
2. G_E においてヒルベルト曲線に沿って可能な限り“UnifyCell”によりセルを合併する。
3. G_E において垂直方向に沿って可能な限り“UnifyCell”によりセルを合併する。
4. G_E において水平方向に沿って可能な限り“UnifyCell”によりセルを合併する。

Algorithm 2 (ヒルベルト走査法を用いた合併→垂直方向合併→水平方向合併)

入力 G_D : 原画像の矩形分割 D に対応する octgrid ($2^n \times 2^n$ サイズ).

出力 G_E : 解像度が低減化された矩形分割 E に対応する octgrid.

方法

1. 初期化 $G_E \leftarrow G_D$.
2. G_E においてヒルベルト曲線に沿って可能な限り“UnifyCell”によりセルを合併する。
3. G_E において水平方向に沿って可能な限り“UnifyCell”によりセルを合併する。
4. G_E において垂直方向に沿って可能な限り“UnifyCell”によりセルを合併する。

またこの2つのアルゴリズムの計算時間は $O((n+m)N)$ である (N は全体のセル数, n は行数, m は列数とする)。

4. 比較

いくつかの画像を例とし、図形の特徴を保ったまま、ヒルベルト走査法を用いたセルの合併による解像度低減を行った2次元画像のセル数と、我々のAlgorithm 1, 2を用いた場合のセル数の比較を行う。

まず単一の図形が存在する画像を例としてセル数の比較を行う。

図5の(a)は一つの三角形が存在する画像(I)に対しヒルベルト走査法を用いたセルの合併の結果で、セルの数は163である。次に(I)に対しAlgorithm 1を適用し(b)を生成する。(b)は垂直方向、水平方向の順で合併を行った結果で、セルの数は59である。さらに(I)に対しAlgorithm 2を適用し(c)を生成する。(c)は水平方向、垂直方向の順で合

併を行った結果で、セルの数は 59 である。(a) に対しての(b)の割合は 36.19%, (c)の割合は 36.19%である。

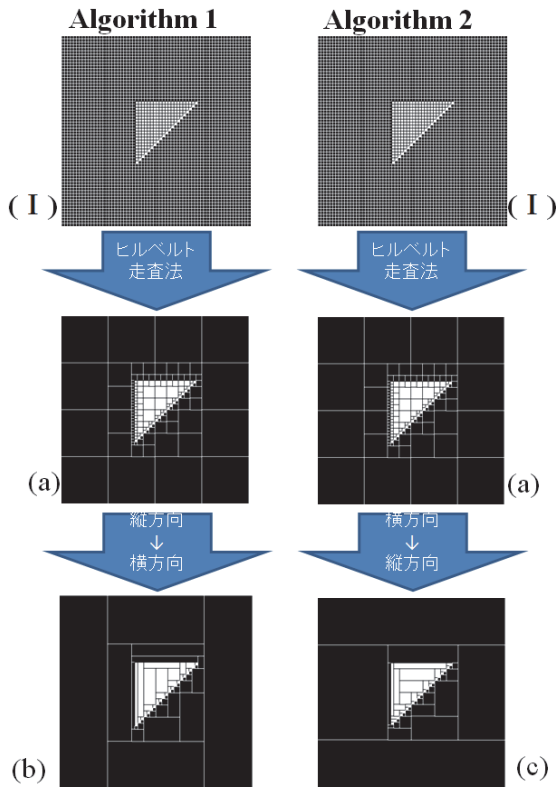


図 5 (I) に対し Algorithm 1 (左), 2 (右) を適用し解像度低減を行った例

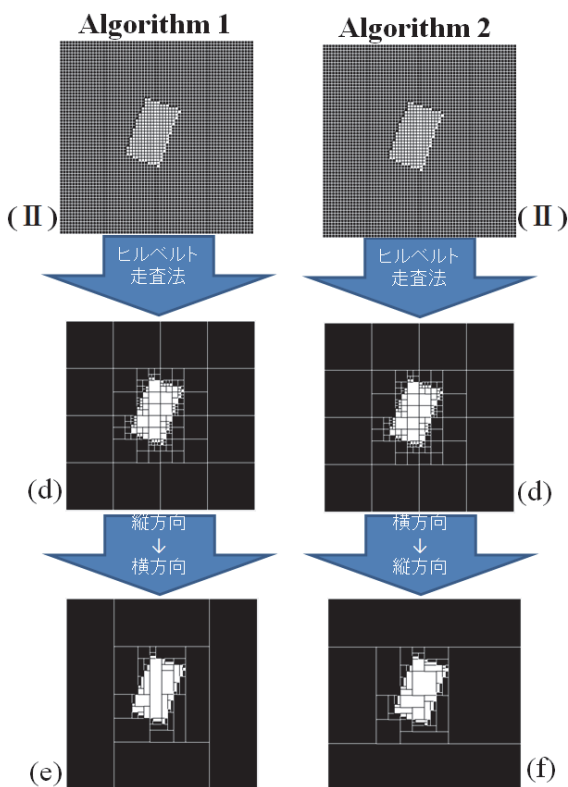


図 6 (II) に対し Algorithm 1 (左), 2 (右) を適用し解像度低減を行った例

図 6の(d)は一つの四角形が存在する画像(II)に対しヒルベルト走査法を用いたセルの合併の結果で、セルの数は 154 である。次に (II) に対し Algorithm 1 を適用し(e)を生成する。(e)は垂直方向、水平方向の順で合併を行った結果で、セルの数は 60 である。さらに (II) に対し Algorithm 2 を適用し(f)を生成する。(f)は水平方向、垂直方向の順で合併を行った結果で、セルの数は 63 である。(d) に対しての(e)の割合は 38.96%, (f)の割合は 40.91%である。

次に、複数の図形が混在する画像を例としてセル数の比較を行う。

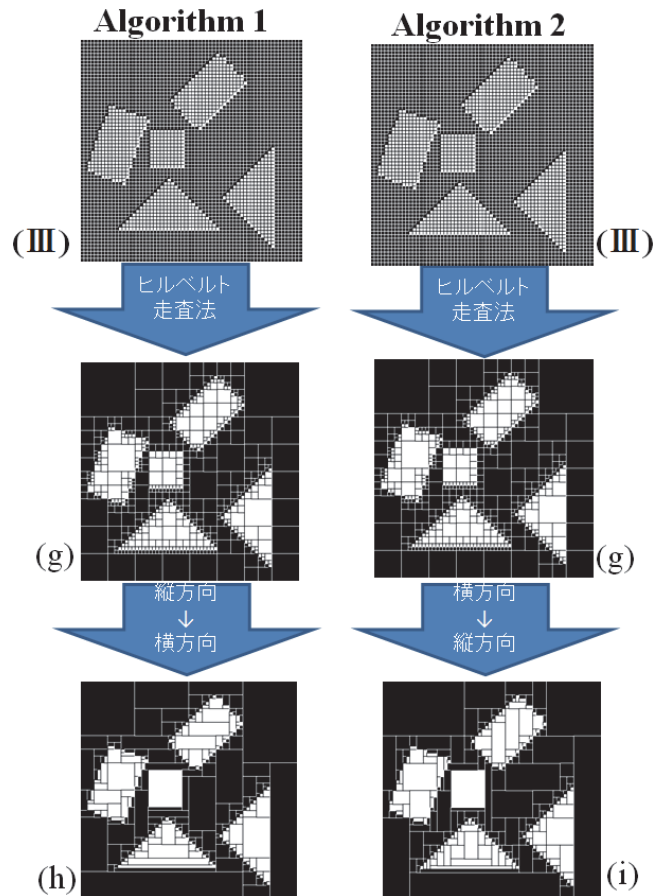


図 7 (III) に対し Algorithm 1 (左), 2 (右) を適用し解像度低減を行った例

図 7の(g)は4つの図形が存在する画像(III)に対しヒルベルト走査法を用いたセルの合併の結果で、セルの数は 619 である。次に (III) に対し Algorithm 1 を適用し(h)を生成する。(h)は垂直方向、水平方向の順で合併を行った結果で、セルの数は 284 である。さらに (III) に対し Algorithm 2 を適用し(i)を生成する。(i)は水平方向、垂直方向の順で合併を行った結果で、セルの数は 276 である。(g) に対しての(h)の割合は 45.88%, (i)の割合は 44.58%である。

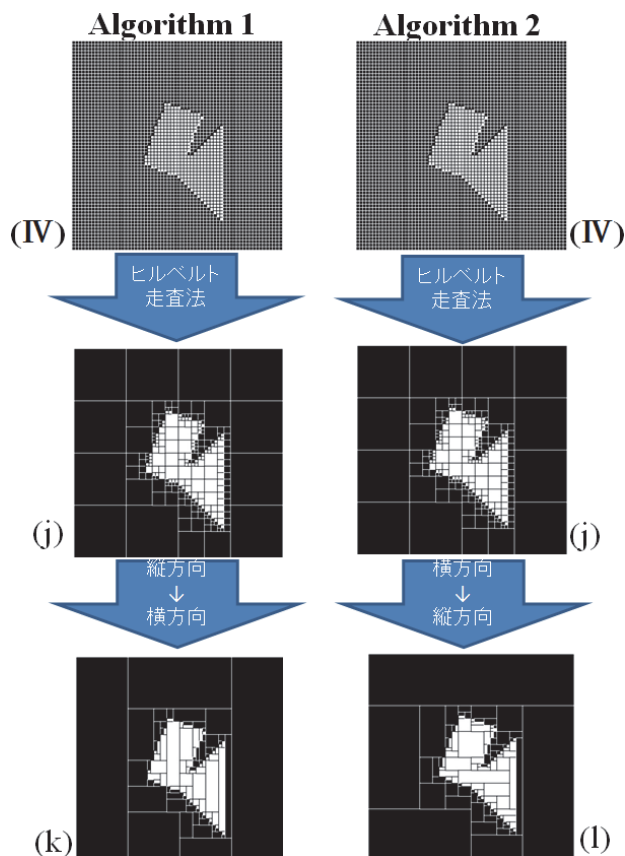


図 8 (IV) に対し Algorithm 1 (左), 2 (右) を適用し解像度低減を行った例

図 8 の(j)は 2 つの重なる図形が存在する画像 (IV) に対しヒルベルト走査法を用いたセルの合併の結果で, セルの数は 244 である. 次に (IV) に対し Algorithm 1 を適用し(k)を生成する. (k)は垂直方向, 水平方向の順で合併を行った結果で, セルの数は 106 である. さらに (IV) に対し Algorithm 2 を適用し(l)を生成する. (l)は水平方向, 垂直方向の順で合併を行った結果で, セルの数は 116 である. (j) に対しての(k)の割合は 43.44%, (l)の割合は 47.54%である.

さらに, 図 9 (V) に 2 値化を行った画像 (VI) を例としてセル数の比較を行う.

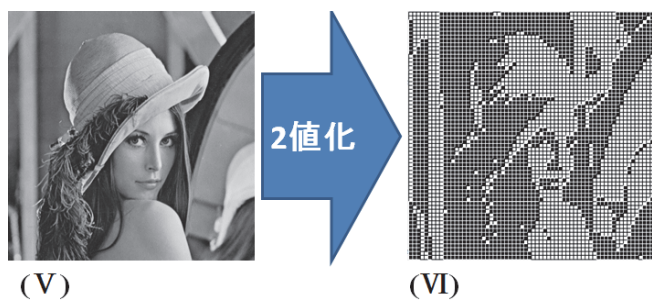


図 9 (V) に対し 2 値化を行った画像 (VI)

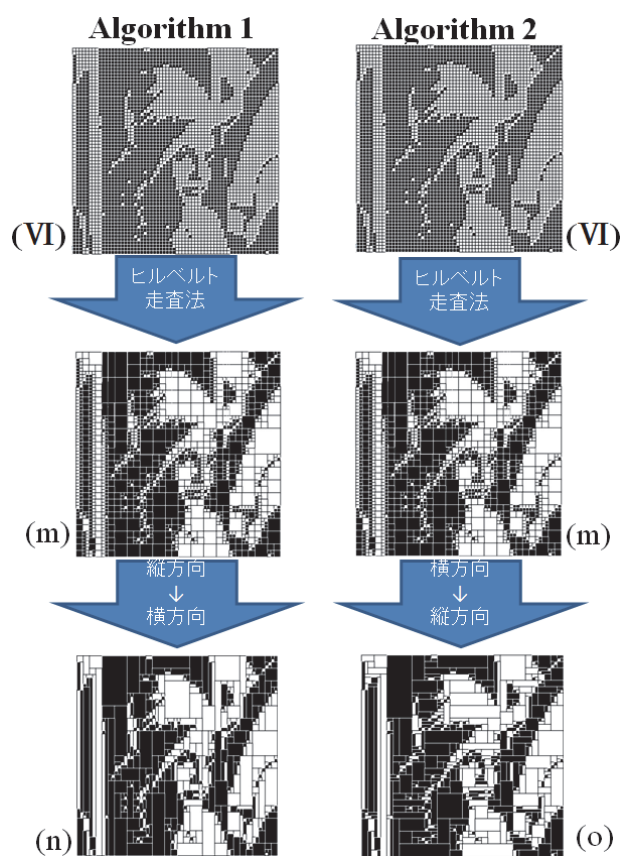


図 10 (VI) に対し Algorithm 1 (左), 2 (右) を適用し解像度低減を行った例

図 10 の(m)は (VI) のヒルベルト走査法によるセルの合併の結果で, セルの数は 1573 である. 次に (VI) に対し Algorithm 1 を適用し(n)を生成する. (n)は垂直方向, 水平方向の順番での合併を行った結果で, セルの数は 578 である. さらに (VI) に対し Algorithm 2 を適用し(o)を生成する. (o)は水平方向, 垂直方向の順番で合併を行った結果で, セルの数は 568 である. (m)に対しての(n)の割合は 36.75%, (o)の割合は 36.11%である.

表 1 は以上の例をまとめたものである.

表 1 ヒルベルト走査法と, Algorithm 1 (上段), 2 (下段) で低減化された 2 次元画像のセル数の比較

原画像	X: ヒルベルト走査法で低減化されたセル数		Y: Algorithm 1 で低減化されたセル数 Y: Algorithm 2 で低減化されたセル数		Y/X
	(a)	(b)	(c)	(d)	
(I) 4096 セル	(a)	163 セル	(b) Algorithm 1	59 セル	36.19%
			(c) Algorithm 2	59 セル	36.19%
(II) 4096 セル	(d)	154 セル	(e) Algorithm 1	60 セル	38.96%
			(f) Algorithm 2	63 セル	40.91%
(III) 4096 セル	(g)	619 セル	(h) Algorithm 1	284 セル	45.88%
			(i) Algorithm 2	276 セル	44.58%
(IV) 4096 セル	(j)	244 セル	(k) Algorithm 1	106 セル	43.44%
			(l) Algorithm 2	116 セル	47.54%
(VI) 4096 セル	(m)	1578 セル	(n) Algorithm 1	578 セル	36.75%
			(o) Algorithm 2	568 セル	36.11%

表より, Algorithm 1, 2 を適用し生成した例のセル数は, ヒルベルト走査法のみを適用し生成した例のセル数の 36.19%~47.54% となり, 大幅なセル数の減少が可能であると示した. また原画像のセル数から Algorithm 1, 2 を適用し生成した例のセル数の割合は, 最大が (n) で 14.11%, 最少が (b), (c) で 1.44% となる.

なお, このアルゴリズムの計算時間は $O((n+m)N)$ であり, 矩形双対グラフで同様の結果を出力するためのアルゴリズムの計算時間 ($O(N^2)$) より速い (N は全体のセル数, n は行数, m は列数とする).

5. まとめ

2次元画像を表現するために octgrid と呼ばれる 8 次格子グラフモデルを紹介した. さらに, 図形の特徴を保ったまま効果的に解像度を減らすためのアルゴリズムを提案した.

また, 例を用いて我々の方法と, 既存の方法であるヒルベルト走査法で低減化された 2次元画像のセル数の比較を示し, 結果として我々の方法が 50%以下にセル数を減少することが可能であると示した.

なお, このアルゴリズムの計算時間は $O((n+m)N)$ であり, 矩形双対グラフで同様の結果を出力するためのアルゴリズムの計算時間 ($O(N^2)$) より速い (N は全体のセル数, n は行数, m は列数とする).

このように, さまざまな多角形や 2 値化された写真に対し我々の方法は矩形双対グラフの方法よりも短い時間で, ヒルベルト走査法よりもセル数を減少させることが出来る (cf., [1]).

今後, 複数図形が混在する画像に対し複数レイヤーに対応させる方法を考察する. また, 図形に対し様々な変形を行う方法を研究し, 実装する (e.g., [7]).

謝辞 本研究にあたり貴重な助言を頂いた東海大学の杉田公生先生, 東洋大学の土田賢省先生, 日本大学の野牧賢志氏, 高加晋司氏, 久保田彬仁氏に深く感謝いたします.

参考文献

- 1) Akagi, G., Anada, K., Koka, S., Nakayama, Y., Nomaki, K. and Yaku, T.: A Resolution Reduction Method for Multi-resolution Terrain Maps, *ACM SIGGRAPH 2012 Posters*, doi= 10.1145/2342896.2342998
- 2) Finkel, R. A. and Bentley, J.L.: Quad Trees: A Data Structure for Retrieval on Composite Keys, *Acta Informatica* 4 (1) pp. 1-9 (1974).
- 3) Hilbert, D.: Über die stetige Abbildung einer Linie auf ein Flächenstück, *Math. Ann.* 38, pp. 459-460 (1891).
- 4) Kamata, S. and Hayashi, Y.: Region-based scanning for image compression, *IEEE International Conference on Image Processing*, pp. 895-898 (2000).
- 5) Kirishima, T., Motohashi, T., Tsuchida, K. and Yaku, T.: Table Processing Based on Attribute Graphs, *Proc. 6th IASTED Conf. SEA*, pp. 317-320 (2004).
- 6) Kozminsky, K. and Kinnen, E.: Rectangular Duals of Planar Graphs, *Networks* 15, pp. 145-157 (1985).
- 7) Shindo, Y., Anada, K., Koka, S. and Yaku, T.: Increasing Ratios of

Cells in Regions Shifted on 2D Images by Raster Data with Heterogeneous Parts, 投稿中.

8) 神藤悠希, 穴田浩一, 夜久竹夫: 平面図形の 8 次格子表現, 第 11 回情報科学技術フォーラム 講演論文集 (*Forum on Information Technology 2012*), pp.133-134 (2012)

9) Shindo, Y., Anada, K. and Yaku, T.: Reduction of Resolution for 2D Images by an Octal Grid Graph Representation Model, 投稿中.

10) Yaku, T.: Representation of Heterogeneous Tessellation Structures by Graphs, *Memoir of WAAP Meetings* 108, 6p, In http://www.waap.gr.jp/waap-memoir/waap108/waap108_02-yaku/011201waap108table-rep-doc.pdf (2001).